IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2019, accepted July 5, 2019, date of publication July 16, 2019, date of current version August 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929152

Birds of a Feather Gel Together: Impact of Team Homogeneity on Software Quality and Team Productivity

NOSHEEN QAMAR[1,2] AND ALI AFZAL MALIK[2], (Senior Member, IEEE)
[1]Department of Computer Science and Information Technology, The University of Lahore, Lahore 54700, Pakistan
[2]Department of Computer Science, National University of Computer and Emerging Sciences, Lahore 54700, Pakistan
Corresponding author: Nosheen Qamar (nosheen.qamar@cs.uol.edu.pk)

ABSTRACT Even though a lot of projects fail due to social issues or personality conflicts, only a small number of empirical studies have been conducted to quantitatively assess the impact of individual personality attributes on the software being developed and the team developing that software. The goal of this paper is to quantify the abstract notion of team homogeneity and to measure its impact on software quality and team productivity. A metric called team homogeneity index (THI) is proposed for this purpose. The six-step process of calculating the THI of a software development team is described and illustrated with the help of an example. Finally, the utility of THI is assessed by conducting a controlled experiment in two different phases of the software development life cycle (SDLC), i.e., implementation and testing. The results reveal that, during the implementation phase, teams with greater THI values were noticeably more productive and produced better quality code. Similarly, during the testing phase, teams with higher values of THI tested more features and wrote better quality test cases. Therefore, the evidence obtained so far suggests that the newly proposed metric, THI, appears to be useful in predicting the quality of software and the productivity of software development teams. Future work includes determining the weights of the five traits using input from the software industry and replication of this empirical study on different phases of SDLC with software practitioners to validate our findings.

INDEX TERMS Five factor model, personality assessment, social aspects of software development, software developer personality traits, software quality, team homogeneity index, team productivity.

I. INTRODUCTION

Software development depends not only on technical activities but also on activities that require interpersonal skills e.g. communication, collaboration, negotiation, leadership, etc. [1]. Software engineers must work in teams and a team with the right people delivers high quality software on time and within budget.

Some previous studies [1]–[5] have explored the importance of human aspects in software development. These studies focused on the relationship between individuals and activities performed by them as members of software engineering teams. According to DeMarco and Lister [6], ''Most software development projects fail because of failures with the team running them''. Human factors are so important that even the most widely used model for effort estimation, COCOMO II takes people attributes (experience and capabilities) into account [7]. Despite their importance, much of the research and practice, however, has focused mainly on technological or process-related factors instead of organizational, social, psychological or personality factors [8].

Evidence from past research [9]–[11] indicates that personality has a great impact on team performance. Personality refers to an individual's persistent and distinguishing characteristics, which makes an individual different from others [12]. The personality of an individual can be assessed using different personality models such as Myers-Briggs Type Indicator (MBTI) [13], Keirsey Temperament Sorter (KTS) [14] and Five Factor Model (FFM) [15] (also known as Big Five Inventory (BFI) [16]).

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Luca Bernardi.

Some models such as MBTI and KTS focus on personality types (i.e. qualitatively distinct categories) while the FFM focuses on personality traits (i.e. characteristics in different dimensions).

MBTI, which is one of the widely used models, uses four bipolar dimensions of personality i.e. Extraversion/Introversion, Sensing/Intuition, Thinking/Feeling and Judging/Perceiving while KTS is based on four temperament types, namely, Artisan, Guardian, Idealist and Rational. FFM measures personality using the following five different dimensions [15]:

Openness**:** This trait suggests that an individual is interested in exploring new things or loves diverse situations.

Conscientiousness: Individuals with high conscientiousness like planning before anything happens instead of being impulsive. They are organized and have a disciplined nature. They plan their tasks and then stick to them with great responsibility.

Extraversion: These people are energetic, social, cheerful, friendly, talkative, and have great communication skills.

Agreeableness: Individuals showing kindness and warm attitude (who are always ready to help others and are sympathetic) are considered to have a high degree of agreeableness.

Neuroticism: People with high neuroticism are inclined to get depressed, frustrated and anxious very easily and frequently. They get worried about minor things.

Many researchers have concluded that personality traits impact project quality (degree to which project meets the requirements) and team productivity (rate of output per unit of input) [17], [18]. However, only a couple of studies have used quantitative aggregation based on a measure of central tendency (i.e. mean) to measure the overall team personality. To the best of our knowledge, no research has so far measured the aggregated score based on a measure of spread. Spread is considered a better representation of a dataset as mean just focuses on the central point to represent the dataset while spread focuses on dispersion or variation across the dataset, for example, mean of two datasets (1, 49) and (24, 26) is the same, ignoring the range or dispersion of dataset. In this research, we have proposed a new metric called Team Homogeneity Index (THI) based on a measure of spread rather than central tendency and have evaluated the impact of THI on software quality and team productivity during the implementation and testing phases of the Software Development Life Cycle (SDLC). Our hypotheses are as follows:

H_{A0}: Greater THI values will have no effect on quality of software.

H_{A1}: Teams with greater THI values will produce better quality software.

H_{B0}: Greater THI values will have no effect on team productivity.

H_{B1}: Teams with greater THI values will be more productive.

These hypotheses are tested by conducting a controlled experiment. The remaining paper is organized as follows. The next section provides a brief review of the relevant literature. The proposed metric (THI) is described in section three. After this, the details of the assessment criteria are given followed by a description of our experiment. Section six presents the results and discusses our main findings. Thereafter, threats to validity are presented in section seven. The last section summarizes the main conclusions and provides suggestions for future work.

II. LITERATURE REVIEW

Many researchers have focused on studying the impact of personality characteristics on software engineering team performance using MBTI and KTS. Rutherfoord [19], for instance, proposed a technique for team formation in software engineering class projects using personality inventories. He claimed that a team with heterogeneous personalities has more skills in solving different problems. Gorla and Lam [20] looked at the personality types of small teams with the aim of finding out the relationship between personality types and team performance. Their results revealed that extroverted practitioners communicated better than introverted individuals and were, therefore, preferable for social interaction tasks. Peslak [31] conducted an empirical study to analyze the impact of personality traits and diversity on project success and team processes. The results revealed that personality has no impact on team processes but it has a positive correlation with project success. Personality diversity was found to have no relationship with project success.

Karn and Cowling [21] analyzed the performance of teams of students by comparing team effectiveness on a yearly basis. Their findings indicated that teams with a variety of personalities brought different ideas which improved the team's productivity. Karn *et al.* [22] evaluated the dynamics of software teams by performing a qualitative analysis for XP projects. Their results indicated that team configuration based on personality types was important for team effectiveness, and teams with high cohesion were found to be more competitive. Capretz [23] conducted a survey on software engineering students by using a personality assessment scale and concluded that variety in personality characteristics leads to better teams and hence improves the quality of products.

A number of researchers have also evaluated the effects of different personality traits on software engineering teams by using the BFI framework. Feldt *et al.* [24] investigated the relationship between personality characteristics and behavior, viewpoints and work preferences of software engineers. For this purpose, they conducted a survey of 47 practitioners working in 10 different software organizations. Their results revealed that conscientiousness (personality trait) has a positive relationship with task preference, acceptance to change and working style. Openness correlates with taking responsibility for the whole project instead of a single task. A recent survey conducted by Yilmaz *et al.* [25] shows that effective team structures support teams with higher emotional stability, agreeableness, extraversion, and conscientiousness.

Different researchers conducted empirical studies to analyze the impact of personality on the performance of a pair in pair programming. The research conducted by Sfetsos *et al.* [26] used a controlled experiment on different pairs. Their results show that pairs with heterogeneous personalities performed better in terms of communication, workability and effectiveness. Walle and Hannay [27] investigated the impact of personality on collaboration in pair programming. Their results revealed that personality correlates positively with collaboration. Diversity in personality traits increases the collaboration and communication between pairs.

Chao and Atli [28] analyzed four personality traits in pair programming. They were not able to find a statistically significant relationship between their selection of personality traits and code quality. Salleh *et al.* [29], [30] assessed the correlation between personality factors like conscientiousness, openness to experience, and neuroticism with the performance of software developers who practice pair programming. The results of their study suggested that conscientiousness does not have a significant impact on performance, although this might be due to the fact that the tasks performed throughout the experiment were short. However, openness to experience was found to have a direct positive correlation with pair productivity.

A couple of studies have used the mean (a measure of central tendency) to aggregate the personality data of software teams. Acuña *et al.* [32] calculated ''Team Personality'' by averaging the scores of each team member for each individual personality factor. They used the mean to aggregate group data. Their results showed that the team with the highest score in agreeableness and conscientiousness personality traits has high job satisfaction. Additionally, they found a positive correlation between extraversion and software product quality. In 2015, Acuña *et al.* [33] replicated their experiment with a greater number of subjects and measured team personality by taking the average of the personalities of all team members. These results revealed that the teams with the highest aggregated score for the agreeableness personality factor have the highest job satisfaction levels. A positive correlation was also found between the extraversion personality factor and software product quality. The four climate factors (i.e. participative safety, team vision, support for innovation, and task orientation) were also found to be positively correlated with job satisfaction in software development teams.

Different approaches have been proposed to improve software team performance using personality traits. Shameem *et al.* [34] proposed a framework to improve team performance by using team members' personality and team climate. Their results reveal that team members with extraversion and conscientiousness personality traits make the best personality-climate combination for effective team formation. Gilal *et al.* [35] suggested a slightly different approach for software team composition. They used three classification techniques e.g. decision tree, logistic regression, and rough sets theory (RST) for software team formation. Based on these techniques, a model was proposed for predicting team performance using three predictors i.e.

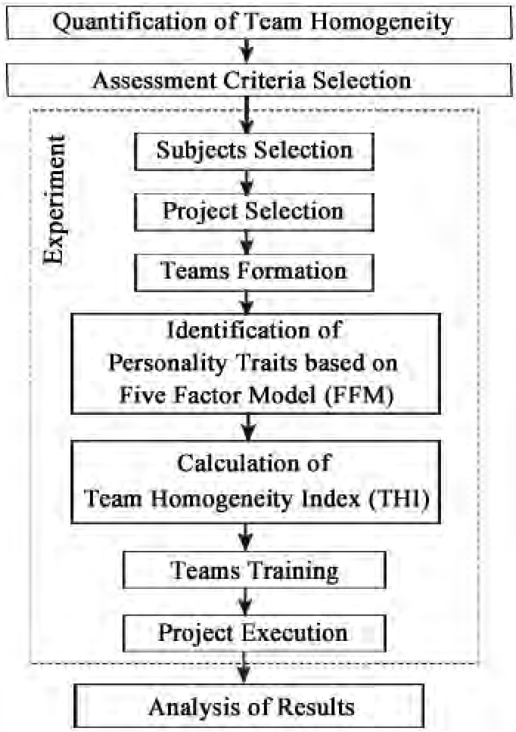


FIGURE 1. Research methodology.

personality type, gender variables, and team role. Results revealed that Johnson algorithm (JA) of RST was the best technique for team composition.

Very recently, in 2018, Poonam and Yasser [36] conducted an empirical study to evaluate the impact of personality traits on pair programming in two different scenarios: first, when the pairs are working together at the same location and, second, when the pairs are working at different locations. Their results reveal that personality traits affect the performance of pairs in the scenario in which pairs are working at different locations.

Table 1 summarizes the past work in this area. So far, to the best of our knowledge, no work has been done to quantify the abstract notion of team homogeneity using a measure of spread. This research attempts to fill this gap.

III. QUANTIFICATION OF TEAM HOMOGENEITY

As shown in Figure 1, quantification of team homogeneity is the first step of our research methodology. Selection of assessment criteria to assess the quality of software and teams' productivity is the second step. This is followed by an experiment to validate our hypotheses.

The process of team homogeneity quantification consists of the following six steps:

***Step 1*:** The first step of this process is the identification of personality characteristics of team members working on some specific software project. The BFI framework (based on FFM), which is one of the most widely used frameworks, was selected for personality assessment. A 50-item five-factor personality test is conducted by using

TABLE 1. State of the art – empirical studies to evaluate the impact of personality characteristics on software teams.

Sr #	Author(s) (Year)	Participants (Research Type)	Contributions	Personality Model	Limitations
1	Rutherfoord (2001)	26 Students (Experiment)	Heterogeneous teams collaborated and cooperated more.	KTS	• Qualitative measurement of team personality
2	Peslak (2006)	55 Undergraduate students (Survey)	Thinking, judging, and extraversion personality traits have a positive correlation with project success.	MBTI	• Small scope (18 teams)
3	Karn et al. (2007)	5 Teams of students (Questionnaire and Experiment)	Teams with homogeneous personalities (high cohesion) work well on projects.	MBTI	• Qualitative measurement • Small team size
4	Sfetsos et al. (2009)	70 Undergraduate students (Experiment)	Pairs with heterogeneous personalities had more communication and collaboration and hence, performed well.	KTS	• Qualitative analysis • Small team size • No evaluation of quality and productivity
5	Acuña et al. (2009)	105 Students (Experiment)	A positive correlation was observed between extraversion and software product quality.	FFM	• Average of all personality traits to take team's personality • Did not check the effect on team's productivity
6	Walle & Hannay (2009)	88 Professionals (Survey)	Personality might affect pair collaboration.	FFM	• Qualitative analysis
7	Salleh et al. (2009)	453 - 1st Semester students (Experiment)	Conscientiousness factor did not affect the performance of students significantly.	FFM	• Qualitative measurement of team formation • Measured single aspect of personality
8	Salleh et al. (2010)		Openness to experience has a positive correlation with students' performance.		
9	Acuña et al. (2015)	136 Students (Experiment)	Positive correlation was observed between team climate factors and job satisfaction.	FFM	• Average of all personality traits used to determine team's personality • Did not check the affect on team's productivity
10	M. Yilmaz et al. (2017)	216 Professionals (Survey)	Effective team structures support teams with higher emotional stability, agreeableness, extraversion, and conscientiousness personality traits.	FFM	• Did not measure the correlation of personality traits with project quality or team productivity.
11	Gilal et al. (2017)	105 Students (Experiment)	A data mining approach (RST) is considered the best technique for team composition by taking personality type, gender variables, and team roles as predictors.	MBTI	• A non-parametric approach • Difficult to implement
12	Poonam and Yasser (2018)	80 Students (Experiment)	Personality traits have a significant impact when pairs work from remote locations.	MBTI	• Small team size

the International Personality Item Pool (IPIP) [37] on all selected team members. A personality score for all five factors (Openness, Conscientiousness, Agreeableness, Extraversion and Neuroticism) is collected for every team member. To bring all values on a scale of 0-1, we used Min-Max normalization [38].

***Step 2*:** After getting the scores of each team member for all five personality traits, the heterogeneity between the score of one team member and each of the rest of the team members for each particular trait is calculated using the following formula [38].

$$\text{Heterogeneity} = |p - q| \tag{1}$$

where

p = score of selected team member for some specific trait

q = score of other team member for that specific trait

Figure 2 shows how heterogeneity for one trait (Openness) for all members of a five-member team is calculated. The four arrows from Member 1 to the rest of the four members show that how much Member 1 is different from Member 2, Member 3, Member 4, and Member 5. Similarly, the three arrows from Member 2 to Member 3, Member 4 and Member 5 show that how much Member 2 is different from Member 3, Member 4 and Member 5 (we do not calculate the heterogeneity again with Member 1 to avoid duplication). The same process is repeated for Member 3 and Member 4. Note that for the last member (Member 5) no processing is needed since this member has already been compared with all other members. By repeating this process for each personality trait, we get a set of five matrices depicting the heterogeneity of every member with other members for each trait. The five matrices obtained for our sample five-member team are shown in Figure 3. Only values below the main diagonal are used to avoid duplications.

***Step 3*:** The next step is the calculation of Overall Heterogeneity for each member-pair (e.g. M1M2, M1M3, M1M4,…,M4M5) using the distance metric (2) [39]. We assign equal weights to each personality trait so the value of W_k is considered as 1 ($W_k = 1$) for every trait. Heterogeneity of team members for each personality trait is

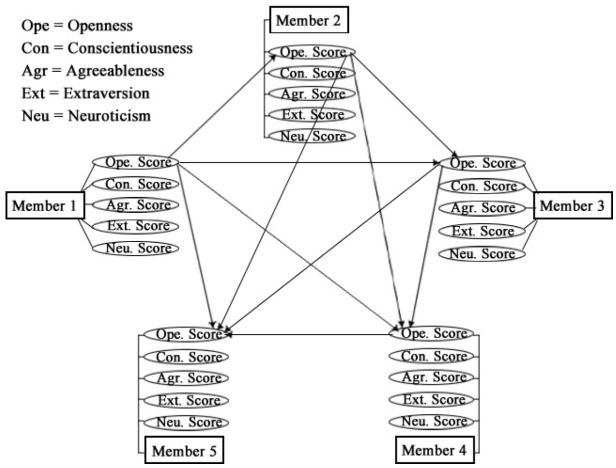
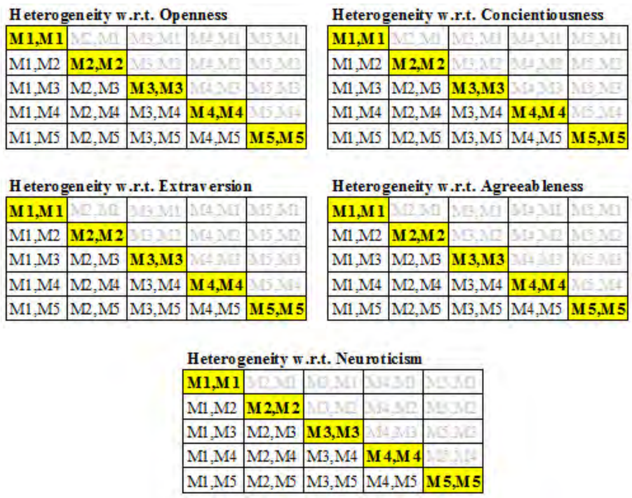


FIGURE 2. Heterogeneity of openness trait.



Heterogeneity w.r.t. Openness

M1,M1	M2,M1	M3,M1	M4,M1	M5,M1
M1,M2	**M2,M2**	M3,M2	M4,M2	M5,M2
M1,M3	M2,M3	**M3,M3**	M4,M3	M5,M3
M1,M4	M2,M4	M3,M4	**M4,M4**	M5,M4
M1,M5	M2,M5	M3,M5	M4,M5	**M5,M5**

Heterogeneity w.r.t. Concientiousness

M1,M1	M2,M1	M3,M1	M4,M1	M5,M1
M1,M2	**M2,M2**	M3,M2	M4,M2	M5,M2
M1,M3	M2,M3	**M3,M3**	M4,M3	M5,M3
M1,M4	M2,M4	M3,M4	**M4,M4**	M5,M4
M1,M5	M2,M5	M3,M5	M4,M5	**M5,M5**

Heterogeneity w.r.t. Extraversion

M1,M1	M2,M1	M3,M1	M4,M1	M5,M1
M1,M2	**M2,M2**	M3,M2	M4,M2	M5,M2
M1,M3	M2,M3	**M3,M3**	M4,M3	M5,M3
M1,M4	M2,M4	M3,M4	**M4,M4**	M5,M4
M1,M5	M2,M5	M3,M5	M4,M5	**M5,M5**

Heterogeneity w.r.t. Agreeableness

M1,M1	M2,M1	M3,M1	M4,M1	M5,M1
M1,M2	**M2,M2**	M3,M2	M4,M2	M5,M2
M1,M3	M2,M3	**M3,M3**	M4,M3	M5,M3
M1,M4	M2,M4	M3,M4	**M4,M4**	M5,M4
M1,M5	M2,M5	M3,M5	M4,M5	**M5,M5**

Heterogeneity w.r.t. Neuroticism

M1,M1	M2,M1	M3,M1	M4,M1	M5,M1
M1,M2	**M2,M2**	M3,M2	M4,M2	M5,M2
M1,M3	M2,M3	**M3,M3**	M4,M3	M5,M3
M1,M4	M2,M4	M3,M4	**M4,M4**	M5,M4
M1,M5	M2,M5	M3,M5	M4,M5	**M5,M5**

FIGURE 3. Heterogeneity of each team member with other team members for each personality trait.

multiplied by its assigned weight and the aggregated heterogeneity value is divided by the number of traits (i.e. 5).

$$\text{Overall Heterogeneity} = \frac{1}{n}\left(\sum_{k=1}^{n} w_k \left|p_k - q_k\right|\right) \tag{2}$$

where

p = score of a selected team member for some specific trait

q = score of other team member for that specific trait

n = number of personality traits (i.e. 5)

w = weight assigned to each personality trait

Step 4: After getting the value of Overall Heterogeneity, the next step is the calculation of Mean using the following formula:

$$\text{Mean} = \frac{(x_1 + x_2 + x_3 + \ldots\ldots + x_m)}{m} \tag{3}$$

where

x1,x2,…,xm = Overall Heterogeneity of member-pairs (e.g. M1M2, M1M3, M1M4,…,M4M5)

TABLE 2. Bugs severity levels & corresponding weights (adapted from [40]).

Severity Level	Description	Weight
Critical	A core functionality of the system fails or the system doesn't work at all.	5
Major	Impacts basic functionality and the system is unable to produce required results.	4
Moderate	Causes the system to generate false, inconsistent, or incomplete results.	3
Minor	Impacts the business, but only in a very few cases.	2
Low	Effects the interface and appearance of the application.	1

m = total number of member-pairs (e.g. 10 for above example of a five-member team)

Step 5: Mean Square Error (MSE) is calculated. To calculate the value of MSE, the absolute difference of Overall Heterogeneity (2) with Mean (3) is calculated and then the sum of all differences is divided by the total number of member-pairs.

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left|Mean - x_i\right| \tag{4}$$

where

x1,x2,…,xm = Overall Heterogeneity of member-pairs (e.g. M1M2, M1M3, M1M4,…,M4M5)

m = total number of member-pairs (e.g. 10 for above example of a five-member team)

Step 6: Finally, Team Homogeneity Index (THI) is calculated by subtracting MSE (calculated using (4)) from 1. THI falls in the range of 0-1 where zero means no similarity at all and 1 means 100% similarity.

$$\text{THI} = 1\text{-MSE} \tag{5}$$

Figure 4 shows how to calculate THI for a sample five-member team using the six-step process described above. It can be seen that each member has a different personality which is captured by a unique set of values corresponding to the five personality traits. In the next step, (1) was used to calculate the Heterogeneity of each team member with the rest of the members for all five traits. To calculate Overall Heterogeneity, (2) was used and then the Mean was calculated by putting all member pair values in (3). After that, MSE was calculated using (4) and finally, MSE was subtracted from 1 to get the THI (using (5)).

IV. ASSESSMENT CRITERIA SELECTION

The project quality and team productivity is assessed for implementation and testing phases based on the criteria given below.

A. ASSESSMENT CRITERIA FOR IMPLEMENTATION PHASE

The quality of implemented projects is measured using four different metrics i.e. weighted sum of bugs (assessed by

1a. Team Members' Scores Obtained From BFI Test

Sr#	T.M. \ P.T.	Ope	Con	Ext	Agr	Neu
1	M1	20	24	21	24	39
2	M2	18	21	33	17	36
3	M3	17	18	25	22	46
4	M4	25	21	30	19	44
5	M5	17	18	35	13	39

=>

1b. Team Members' Scores After Normalization

Sr#	T.M. \ P.T.	Ope	Con	Ext	Agr	Neu
1	M1	0.375	1.000	0.000	1.000	0.300
2	M2	0.125	0.500	0.857	0.364	0.000
3	M3	0.000	0.000	0.286	0.818	1.000
4	M4	1.000	0.500	0.643	0.545	0.800
5	M5	0.000	0.000	1.000	0.000	0.300

*Ope= Openness, Con= Conscientiousness, Ext=Extraversion, Agr=Agreeableness, Neu=Neuroticism
T.M.= Team Member, P.T.= Personality Trait

2a. Heterogeneity w.r.t. Openness

0.250			
0.375	0.125		
0.625	0.875	1.000	
0.375	0.125	0.000	1.000

2b. Heterogeneity w.r.t. Concientiousness

0.500			
1.000	0.500		
0.500	0.000	0.500	
1.000	0.500	0.000	0.500

2c. Heterogeneity w.r.t. Extraversion

0.857			
0.286	0.571		
0.643	0.214	0.357	
1.000	0.143	0.714	0.357

2d. Heterogeneity w.r.t. Agreeableness

0.636			
0.182	0.455		
0.455	0.182	0.273	
1.000	0.364	0.818	0.545

2e. Heterogeneity w.r.t. Neuroticism

0.300			
0.700	1.000		
0.500	0.800	0.200	
0.000	0.300	0.700	0.500

3. Overall Heterogeneity

0.509			
0.509	**0.530**		
0.544	**0.414**	**0.466**	
0.675	**0.286**	**0.446**	**0.581**

4 MEAN = (0.509+0.509+0.544+0.675+0.530+0.414+0.286+0.466+0.446+0.581)/10 = **0.496**

5 MSE = (|0.496-0.509|+|0.496-0.509|+|0.496-0.544|+|0.496-0.675|+|0.496-0.530|+|0.496-0.414|
+|0.496-0.286|+|0.496-0.466|+|0.496-0.446|+|0.496-0.581|)/10 = **0.074**

6 THI = 1-0.074 = **0.926**

FIGURE 4. THI calculation example.

looking at the number and severity level of bugs as shown in Table 2), defect density, cyclomatic complexity [41], and maintainability index [42]. The latter two metrics are calculated automatically using the PhpMetrics tool [43]. The formulas for calculating each of these four metrics are given below:

$$\text{Weighted Sum of Bugs} = 5(A)+4(B)+3(C)+2(D)+1(E) \quad (6)$$

where
A = number of critical bugs
B = number of major bugs
C = number of moderate bugs
D = number of minor bugs
E = number of low bugs

$$\text{Defect Density} = \text{Weighted Sum of Bugs}/\text{LOC} \quad (7)$$

where
LOC = (source) lines of code

$$\text{Cyclomatic Complexity} = E - N + 2^{*}P \quad (8)$$

where
E = number of edges in the flow graph
N = number of nodes in the flow graph
P = number of nodes that have exit points

$$MI = 171 - 5.2^{*}\ln(V) - 0.23^{*}(G) - 16.2^{*}\ln(LOC) \quad (9)$$

where
MI = maintainability index
V = halstead volume
G = cyclomatic complexity

The productivity of software implementation teams is assessed using metrics calculated using the following formulas:

$$\text{Project Completeness} = (100(A) + 75(B) + 50(C) + 25(D) + 0(E))/100 \quad (10)$$

where
A = number of requirements that have been implemented completely

B = number of requirements that have been implemented 75%

C = number of requirements that have been implemented 50%

D = number of requirements that have been implemented 25%

E = number of requirements that have not been implemented at all.

$$\text{Productivity} = \text{Project Completeness/Effort} \quad (11)$$

where

Effort = total Person Hours (P.H.) taken to implement the project

$$\text{FP Productivity} = (\text{LOC}/67)/\text{Effort} \quad (12)$$

where

FP = Function Points (67 LOC of PHP = 1 FP [44])

B. ASSESSMENT CRITERIA FOR TESTING PHASE

The quality of the work done by testing teams is assessed using the following formulas:

$$\text{Defects Uncovered} = \text{Failed Test Cases} \quad (13)$$

$$\text{Architectural Coverage}(\%) = (\text{Features Tested}/\text{Total Features}) * 100 \quad (14)$$

$$\text{Test Case Conformity } (\%) = (\text{CTCA}/\text{TTCA}) * 100 \quad (15)$$

where

CTCA = Correct Test Case Attributes = test case attributes conforming to the given template and guidelines

TTCA = Total Test Case Attributes = total test case attributes provided by a team

Productivity of software testing teams is assessed using the following formula:

$$\text{Productivity} = \text{Features Tested/Effort} \quad (16)$$

where

Effort = total Person Hours (P.H.) taken to test the project

C. ANALYSIS OF STATISTICAL SIGNIFICANCE

In order to test the null hypotheses (H_{A0} and H_{B0}), a one-way Analysis Of Variance (ANOVA) [45] test is carried out to analyze the significant difference in the scores of software quality and teams' productivity (dependent variables) verses THI (independent variable) between the teams. The ANOVA produces the significance of F (*p*-value) which represents the sufficient evidence or level of confidence to accept or reject the null hypotheses. SPSS [46] software is used to conduct this ANOVA test.

V. EXPERIMENT

In order to ascertain the utility of this newly proposed metric–THI–an experiment was designed. Different steps of this experiment are depicted in Figure 1 and described in this section.

TABLE 3. Experiment details.

Attributes	Implementation Phase	Testing Phase
No. of participants	90	45
Course	Web Engineering	Software Testing
Semester	6th	8th
No. of Teams	18	9
Members in Each Team	5	5
Task Duration	4 Weeks	2 Working Days
Overall Experiment Duration	8 Weeks	4 Weeks

A. SUBJECTS SELECTION

The experiment used 6th and 8th semester BS (Computer Science) students of an emerging private university in Lahore during the academic year 2017-2018. This experiment was conducted in two different phases of the SDLC i.e. implementation and testing. Table 3 summarizes the important details of this experiment. A total of 135 students participated in this experiment. 117 were male and 8 were female students. 90 students studying the "Web Engineering" course participated in the implementation phase and 45 students studying the "Software Testing" course participated in the testing phase.

Given that this experiment was meant to serve as a proof of concept, the choice of an academic environment is justified [47], [48]. Sjoeberg *et al.* [49] have reported that conducting experiments in an industrial environment increases realism but at more cost along with different internal and conclusion validity threats which take the study away from its actual goal. According to Hornbaek [50], "in itself having students participate in an experiment may not matter to a study" (page: 27).

B. PROJECT SELECTION

A web-based E-commerce project was selected for the implementation phase since the participants were enrolled in the "Web Engineering" course. For the testing phase, this same project (completely implemented by the first author using PHP) was given to the participants enrolled in the "Software Testing" course for testing purposes.

C. TEAMS FORMATION

Teams were formed by mixing students belonging to the following three buckets of CGPA:

Bucket 1: 3.00 – 4.00

Bucket 2: 2.50 – 2.99

Bucket 3: 2.00 – 2.49

In each team, one member was selected from bucket 1 and two members each from buckets 2 and 3. These members were selected randomly using the random.org website [51]. The average CGPA for each team was between 2.4 – 2.6.

Since there were very few female participants (8 out of 135), it was ensured that no more than one female student is present in a single team. We also ensured that the average grade in previously studied programming courses for each team was at least B. Furthermore, in order to minimize the impact of past experience, we ensured that there was no more than one student having practical experience (in the form of internship or freelancing) in each team.

D. IDENTIFICATION OF PERSONALITY TRAITS BASED ON FIVE FACTOR MODEL (FFM)

A 50-item five-factor personality test based on the International Personality Item Pool (IPIP) [37] was taken by all subjects. A training session of 30 minutes duration was conducted before this test to make students understand the importance and material of this test. The important terms which were to be used in this test were explained in detail during this training session. The personality test was conducted during the lab sessions of the respective course using an online application [52] made by the first author.

E. CALCULATION OF TEAM HOMOGENEITY INDEX

After the collection of all information regarding the personality traits, the Team Homogeneity Index (THI) was automatically calculated (using an online tool [53] made by the first author) for each team using the process discussed in Section 3.

F. TEAMS TRAINING

A total of six training sessions were conducted. As shown in Table 4, four of these six sessions were used for training the implementation teams: two 30-minutes sessions before conducting personality assessment test and two (3 hours each) training sessions before starting the actual experiment. Since participants were split into two equal groups of 45 members, each group attended only two training sessions-one for personality assessment and one for software implementation. The agenda of software implementation training session was as follows:

- Detailed introduction of SRS document (E-commerce Project)
- Software implementation guidelines
  - One member will perform the role of Team Lead and the rest will serve as Developers/Unit Testers.
  - The Front End of the application needs to be developed using HTML 4/5 and CSS 3 using the Bootstrap framework.
  - PHP will be used as the Server End programming language and Javascript will be used as Client End scripting language.
  - The database will be built using Mysql.
  - Teams will follow the Waterfall process.
  - Teams will use the MVC (Model View Controller) architectural pattern to design the project.
  - Teams will have four weeks to complete their project.
- Demonstration of "TimeKeeper" [54] tool for logging time.

Two sessions were used for software testing teams: one 30-minute session before conducting the personality assessment test and a 3-hours session before starting the actual experiment. The agenda of software testing training session was as follows:

- Detailed introduction of SRS document (E-commerce Project)
- Software testing guidelines
  - One member will perform the role of Team Lead and the rest will work as team members.
  - Teams will perform Black Box [40] testing only.
  - Test cases will be written using the provided template [55].
  - Equivalence Class Partitioning [40] approach will be used to create test cases.
  - Test cases will be based on the provided SRS document.
  - Teams will have two working days to complete their project.
- Demonstration of "TimeKeeper" [54] tool for logging time.

G. PROJECT EXECUTION

After the completion of training sessions, the E-commerce Project was assigned to "Web Engineering" students for implementation and to "Software Testing" students for testing. The maximum time given for implementation of the project was four weeks. For testing this project, only two working days were given.

VI. RESULTS AND DISCUSSION

A. RESULTS OF IMPLEMENTATION PHASE

Table 5 shows the details regarding the quality of projects implemented by different teams. Columns 4-8 depict the number of bugs of different severity levels while the ninth column shows the weighted sum of bugs calculated using (6). The values of Defect Density, Cyclomatic Complexity, and Maintainability Index (MI) are given in the last three columns.

Figures 5A and 5B display scatter plots with accompanying trend-lines that show the impact of THI on weighted sum of bugs and defect density of projects implemented. The downward slopping trend-lines make it clear that the teams with greater THI values implemented the project with fewer weighted bugs and lower defect density. The relationships between THI and cyclomatic complexity and THI and maintainability index are shown in Figures 5C and 5D, respectively. Though R^2 values are not very strong, the overall directions of the relationships seem promising. Higher values of THI seem to be associated with lower complexity and better maintainability. Thus, Figure 5 somewhat supports our second hypothesis (H_{A1}).

Table 6 provides the details of productivity calculations for the software implementation teams. Columns 4-8 contains information related to task completeness i.e. number of requirements that were implemented 100%, 75% and so on. The values for project completeness in column 9 were

TABLE 4. Overview of training sessions.

Training Session	Group	Purpose	Course	Participants	Duration	Training Agenda
Implementation Phase						
Session 1	Group 1	Personality Assessment	Web Engineering	45	30 min	Introduction to personality test, Explanation of questions
Session 2	Group 2					
Session 3	Group 1	Software Implementation	Web Engineering	45	3 hours	SRS document explanation, Software implementation guidelines, TimeKeeper tool demonstration
Session 4	Group 2					
Testing Phase						
Session 5	Group 1	Personality Assessment	Software Testing	45	30 min	Introduction to personality test, Explanation of questions
Session 6	Group 1	Software Testing	Software Testing	45	3 hours	SRS document explanation, Software testing guidelines, TimeKeeper tool demonstration

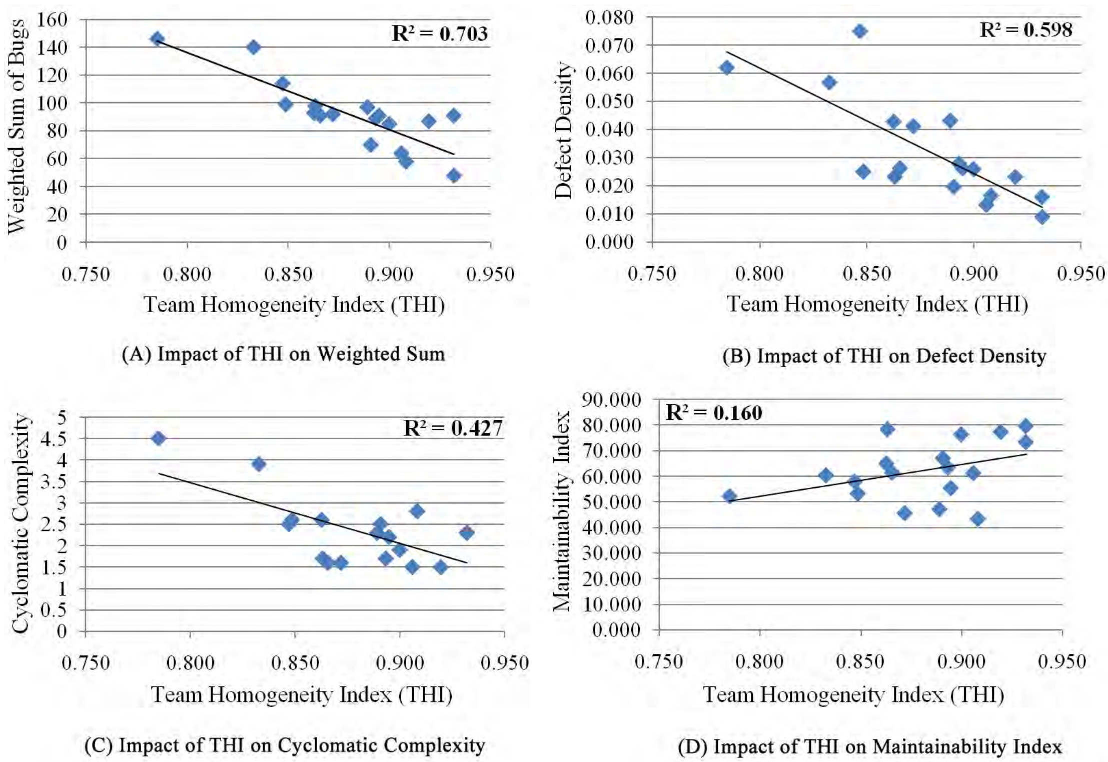


FIGURE 5. Impact of THI on quality of implemented software.

calculated using (10). Column 11 shows the productivity of implementation teams which was obtained using (11). The last column shows the values of FP productivity which were calculated using (12).

Figure 6 shows scatter plots with accompanying trend-lines that depict the relationships between THI and productivity and THI and FP productivity, respectively. These upward sloping trend-lines indicate that teams with higher values of THI are more productive. This lends support to our last hypothesis (H_{B1}).

B. RESULTS OF TESTING PHASE

The results of testing phase also appear promising. Table 7 provides the details of all the attributes used for assessing testing teams' quality. Column 4 shows the defects uncovered by each team. Architectural coverage (calculated using (14)) can be seen in column 5. Columns 6-8 represent correct test case attributes, incorrect test case attributes and total test case attributes, respectively. Column 9 contains the values of conformity to test case template (obtained using (15)).

Figure 7 shows the relationship between THI and quality of work done by testing teams. Figure 7A shows the impact of THI on the number of defects uncovered by testing teams. It is clear from this figure that the teams with higher THI seem to have identified more bugs. Figure 7B depicts the impact of THI on the architectural coverage achieved by testing teams. Teams with greater THI seem to achieve better architectural converge as compared to teams with lower THI. Quality of the test case document was also assessed in terms of conformity

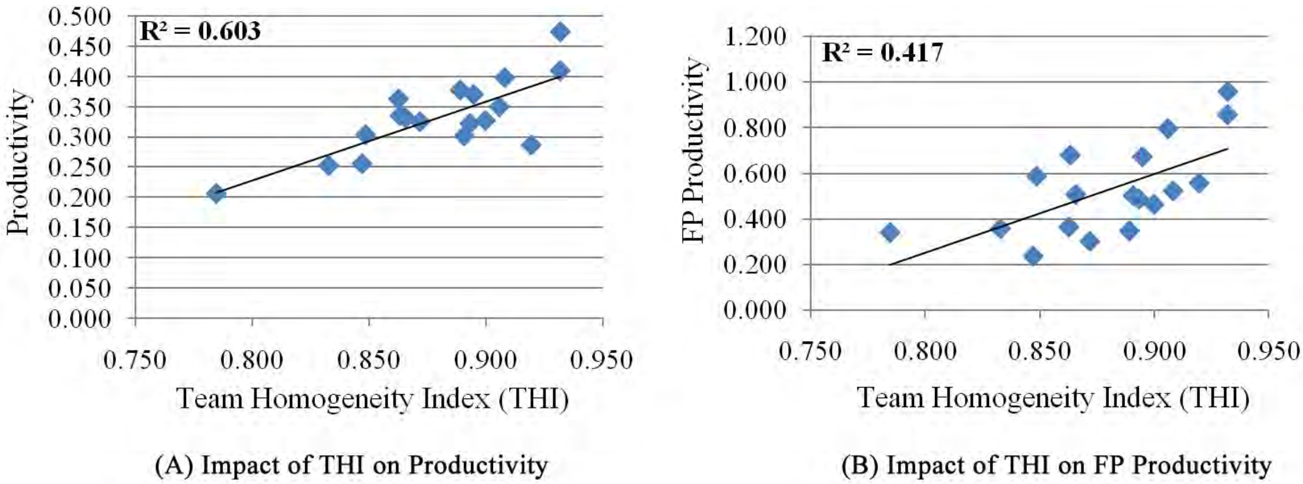


FIGURE 6. Impact of THI on productivity of implementation teams.

TABLE 5. Implementation teams' quality.

Sr#	Team	THI	Number of Bugs					Weighted Sum of Bugs	Defect Density	Cyclomatic Complexity	MI
			Critical (5)	Major (4)	Moderate (3)	Minor (2)	Low (1)				
1	Team 1	0.932	2	4	4	2	6	48	0.009	2.3	73.330
2	Team 2	0.863	7	12	2	3	3	98	0.021	1.7	70.235
3	Team 3	0.906	7	5	2	1	1	64	0.013	1.5	61.235
4	Team 4	0.849	8	10	4	2	3	99	0.025	2.6	53.255
5	Team 5	0.893	7	9	2	2	8	89	0.028	1.7	63.425
6	Team 6	0.891	3	8	3	4	6	70	0.032	2.5	67.036
7	Team 7	0.872	2	13	5	4	7	92	0.041	1.6	45.651
8	Team 8	0.908	3	6	3	3	4	58	0.016	2.8	43.392
9	Team 9	0.833	12	14	6	2	2	140	0.057	3.9	60.424
10	Team 10	0.889	3	12	8	4	2	97	0.043	2.3	47.235
11	Team 11	0.863	6	11	3	2	6	93	0.043	2.6	64.936
12	Team 12	0.895	5	9	6	5	2	91	0.026	2.2	55.456
13	Team 13	0.900	4	7	7	6	4	85	0.037	1.9	78.253
14	Team 14	0.866	3	10	8	4	4	91	0.026	1.6	61.436
15	Team 15	0.919	9	7	2	3	2	87	0.023	1.5	77.219
16	Team 16	0.847	11	9	5	2	4	114	0.075	2.5	57.945
17	Team 17	0.932	4	10	7	4	2	91	0.019	2.3	79.548
18	Team 18	0.785	17	9	5	4	2	146	0.062	4.5	48.235

to the given template and provided guidelines. Figure 7C shows that teams with higher values of THI followed the template more strictly.

The positive correlation between each of defects uncovered, architectural coverage and test case conformity and THI supports our second hypothesis (H_{A1}) in the testing phase as well.

Table 8 provides the details of all the attributes required to calculate the productivity of testing teams. Column four contains the number of test cases written by each team while the total number of tested features appears in column five. Productivity (calculated using (16)) is shown in the last column.

Figure 8 shows the relationship between THI and productivity of testing teams. It is evident from this figure that teams with higher values of THI appear to be more productive during the testing phase as well. In fact, productivity and THI are more strongly correlated ($R^2 = 0.654$) during the testing phase as compared to the implementation phase ($R^2 = 0.603$) which goes in favor of our last hypothesis (H_{B1}).

C. HYPOTHESES TESTING

The null hypotheses (H_{A0} and H_{B0}) were tested using one-way analysis of variance (ANOVA). A total of 10 ANOVA tests, one for each productivity and quality factor (assessment criterion) were conducted. Table 9 provides a summary of the results of these 10 tests. It can be seen in this table that for the implementation phase, the p values for productivity and FP productivity are less than 0.05. This shows that there is sufficient evidence to reject the null hypothesis H_{A0} and accept H_{A1}. Similarly, *p* values for almost all quality factors (i.e. weighted sum of bugs, defects density, cyclomatic complexity) are less than 0.05. This indicates that there is sufficient evidence to accept H_{B1} and reject H_{B0} for these quality factors. The *p* value for only one quality factor i.e. maintainability index is more than 0.05. Thus, for maintainability index, evidence is not sufficient to reject H_{B0}.

For the testing phase, the *p* value for productivity is 0.00824 (< 0.05). This indicates that there is ample evidence

TABLE 6. Implementation teams' productivity.

Sr#	Team	THI	Task Completion (Total Requirements 51)					Project Comp	Effort (Total P.H.)	Prod	LOC	FP	FP Prod
			100%	75%	50%	25%	0%						
1	Team 1	0.932	30	10	4	2	2	40.000	84.380	0.474	5405	80.672	0.956
2	Team 2	0.863	18	11	5	10	7	31.250	93.110	0.336	4230	63.134	0.678
3	Team 3	0.906	12	20	8	4	7	32.000	91.410	0.350	4860	72.537	0.794
4	Team 4	0.849	21	7	3	12	8	30.750	101.170	0.304	3963	59.149	0.585
5	Team 5	0.893	17	13	6	8	7	31.750	98.400	0.323	3194	47.672	0.484
6	Team 6	0.891	9	20	14	5	3	32.250	106.730	0.302	3575	53.358	0.500
7	Team 7	0.872	12	25	10	2	2	36.250	111.380	0.325	2235	33.358	0.299
8	Team 8	0.908	30	9	5	4	3	40.250	101.050	0.398	3526	52.627	0.521
9	Team 9	0.833	14	12	6	1	12	26.250	103.660	0.253	2467	36.821	0.355
10	Team 10	0.889	18	17	10	4	3	36.750	97.250	0.378	2251	33.597	0.345
11	Team 11	0.863	12	20	9	4	6	32.500	89.500	0.363	2174	32.448	0.363
12	Team 12	0.895	3	25	10	8	5	28.750	77.580	0.371	3483	51.985	0.670
13	Team 13	0.900	13	22	9	3	4	34.750	106.210	0.327	3283	49.000	0.461
14	Team 14	0.866	12	22	8	6	3	34.000	102.550	0.332	3470	51.791	0.505
15	Team 15	0.919	13	11	13	5	9	29.000	101.120	0.287	3765	56.194	0.556
16	Team 16	0.847	15	2	10	13	11	24.750	96.650	0.256	1521	22.701	0.235
17	Team 17	0.932	35	3	5	4	4	40.750	99.480	0.410	5699	85.060	0.855
18	Team 18	0.785	10	6	10	8	17	21.500	104.150	0.206	2355	35.149	0.337
* Comp = Completeness, Prod = Productivity, P.H.=Person Hours													

TABLE 7. Testing teams' quality.

Sr#	Team	THI	Defects Uncovered	Architectural Coverage	CTCA	ITCA	TTCA	Test Case Conformity
1	Team 1	0.808	5	37.255	293	59	352	83.239
2	Team 2	0.850	6	50.980	262	49	311	84.244
3	Team 3	0.905	8	60.784	603	57	660	91.364
4	Team 4	0.873	6	35.294	325	50	374	86.898
5	Team 5	0.827	4	27.451	151	25	176	85.795
6	Team 6	0.883	4	50.980	517	55	572	90.385
7	Team 7	0.799	2	19.608	124	20	154	80.519
8	Team 8	0.825	5	29.412	211	31	242	87.190
9	Team 9	0.896	7	47.059	315	26	341	92.375
*CTCA= Correct Test Case Attributes, ITCA= Incorrect Test Case Attributes, TTCA= Total Test Case Attributes								

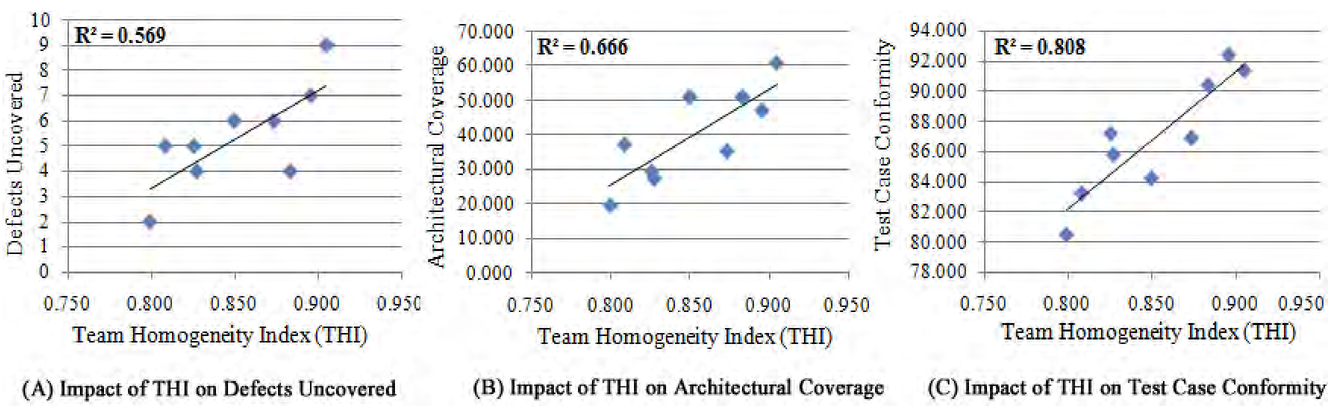


FIGURE 7. Impact of THI on quality achieved by testing teams.

to reject the null hypothesis H_{A0} and accept the H_{A1}. In case of testing teams' quality factors (i.e. architectural coverage, defects uncovered and test case conformity) also the *p* values are less than 0.05 for all three factors providing sufficient support to accept H_{B1} and reject H_{B0}.

VII. THREATS TO VALIDITY

Although the results of our experiment seem to be in favor of our hypotheses, some factors cannot be ignored while interpreting the results. Firstly, the subjects' learning ability, intelligence, and individual interest in programming can

TABLE 8. Testing teams' productivity.

Sr#	Team	THI	No. of Test Cases	Features Tested	Effort (Total P.H.)	Productivity
1	Team 1	0.808	32	19	12.600	1.508
2	Team 2	0.850	31	26	13.700	1.898
3	Team 3	0.905	60	31	10.500	2.952
4	Team 4	0.873	34	18	11.600	1.552
5	Team 5	0.827	18	14	14.730	0.950
6	Team 6	0.883	52	26	12.250	2.122
7	Team 7	0.799	14	10	15.300	0.654
8	Team 8	0.825	22	15	13.200	1.136
9	Team 9	0.896	31	24	14.190	1.691

TABLE 9. ANOVA results.

Sr#	Dependent Variables		Degrees of Freedom (df)	Sum of Squares (SS)	Mean Squares (MS)	F	Sig. (*p*-value)
Implementation Phase							
1	Productivity	Productivity	1	0.04048	0.04048	24.33344	0.00015
2		FP Productivity	1	0.28405	0.28405	11.47644	0.00376
3	Quality	Weighted Sum of Bugs	1	7220.88637	7220.88637	38.00950	0.00001
4		Defect Density	1	0.00332	0.00332	23.79943	0.00017
5		Cyclomatic Complexity	1	4.72494	4.72494	11.97124	0.00323
6		Maintainability Index	1	359.06080	359.06080	3.06015	0.09939
Testing Phase							
7	Productivity		1	2.45043	2.45043	13.28238	0.00824
8	Quality	Architectural Coverage	1	958.51147	958.51147	13.99702	0.00725
9		Defects Uncovered	1	18.22234	18.22234	9.25821	0.01877
10		Test Case Conformity	1	100.68977	100.68977	29.56867	0.00097

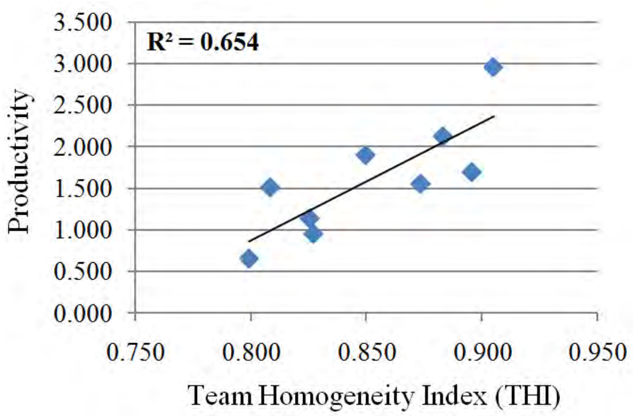


FIGURE 8. Impact of THI on productivity of testing teams.

affect the teams' productivity and software quality. Secondly, the degree of friendship can influence the results as the members of a team are class fellows and some may have better bonding. Furthermore, the level of previous experience and understanding of similar real-life projects can be a confounding factor.

We tried to minimize these threats by making three buckets of CGPA and selected members from each bucket randomly. We also made sure that the teams' average CGPA fall in a specific range to produce teams with an equal level of intelligence, learning ability, and competency. This selection process also helped us to avoid the impact of bonding or friendship on teams' performance. Furthermore, we made sure that there would be no more than one member having some previous experience or understanding of projects was the part of a team to produce teams with uniform capabilities.

VIII. CONCLUSIONS AND FUTURE WORK

The primary aim of this research was the quantification of the abstract concept of team homogeneity. For this purpose, the THI metric was introduced. The impact of THI on teams' productivity and software quality during implementation and testing phases was evaluated using a carefully designed controlled experiment. The results of this experiment reveal that birds of a feather do, indeed, gel together i.e. teams with greater THI are more productive and produce better quality software (as compared to teams with lower values of THI) during both SDLC phases.

This empirical study is just a first step towards evaluating the impact of THI on team productivity and software quality. Future work includes determining the weights of the five traits using input from software industry. We also need to replicate this empirical study on software practitioners to evaluate whether or not the correlation found between THI and team productivity and software quality in an academic setting holds for the industrial environment as well. Last, but not the least, we also plan to look at other phases of the SDLC such as requirements engineering, analysis, and design.

REFERENCES

[1] S. T. Acuña and N. Juristo, ‘‘Assigning people to roles in software projects,’’ *Softw. Pract. Exper.*, vol. 34, no. 7, pp. 675–696, Jun. 2004.

[2] R. T. Turley and J. M. Bieman, ‘‘Competencies of exceptional and nonexceptional software engineers,’’ *J. Syst. Softw.*, vol. 28, no. 1, pp. 19–38, Jan. 1995.

[3] J. L. Wynekoop and D. B. Walz, ‘‘Investigating traits of top performing software developers,’’ *Inf. Technol. People*, vol. 13, no. 3, pp. 186–195, Sep. 2000.

[4] E. Moore, ‘‘Personality characteristics of information systems professionals,’’ in *Proc. Conf. Manage. Inf. Syst. Personnel (SIGCPR)*, Mar. 1991, pp. 140–155.

[5] R. Colomo-Palacios, C. Casado-Lumbreras, pp. Soto–Acosta, S. Misra, and F. García-Peñalvo, ‘‘Analyzing human resource management practices within the GSD context,’’ *J. Global Inf. Technol. Manage.*, vol. 15, no. 3, pp. 30-54, Jul. 2012.

[6] T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*. New York, NY, USA: Dorset House, 1999.

[7] B. W. Boehm, *Software Cost Estimation With COCOMO II*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[8] P. Lenberg, R. Feldt, and L. G. Wallgren, ‘‘Behavioral software engineering: A definition and systematic literature review,’’ *J. Syst. Softw.*, vol. 107, pp. 15–37, Sep. 2015.

[9] J. H. Bradley and F. J. Hebert, ‘‘The effect of personality type on team performance,’’ *J. Manage. Develop.*, vol. 16, no. 5, pp. 337–353, 1997.

[10] L. F. Capretz and F. Ahmed, ‘‘Making sense of software development and personality types,’’ *IT Prof.*, vol. 12, no. 1, pp. 6–13, 2010.

[11] S. T. Acuña, M. Gómez, and N. Juristo, ‘‘How do personality, team processes and task characteristics relate to job satisfaction and software quality?’’ *Inf. Softw. Technol.*, vol. 51, no. 3, pp. 627–639, Mar. 2009.

[12] L. A. Pervin, *Personality: Theory, Assessment and Research*. Hoboken, NJ, USA: Wiley, 1989.

[13] I. B. Myers, H. M. McCaulley, N. L. Quenk, and A. L. Hammer, *MBTI Manual: A Guide to the Development and Use of the Myers Briggs Type Indicator*. Soviet Union, USA: Consulting Psychologists Press, 1998.

[14] D. Keirsey and M. M. Bates, *Please Understand Me*. San Diego, CA, USA: Prometheus Nemesis, 1984.

[15] R. R. McCrae and O. P. John, ‘‘An introduction to the five-factor model and its applications,’’ *J. Pers.*, vol. 60, no. 2, pp. 175–215, 1992.

[16] O. P. John, E. M. Donahue, and R. L. Kentle, ‘‘The big five inventory versions 4a and 54,’’ M.S. Thesis, Dept. Psychol., Univ. California, Berkeley, CA, USA, 1991.

[17] S. Cruz, F. Q. B. Silva, and L. F. Capretz, ‘‘Forty years of research on personality in software engineering: A mapping study,’’ *Comput. Hum. Behav.*, vol. 46, no. 1, pp. 94–113, May 2015.

[18] A. Barroso, J. Silva, M. Soares, and R. Nascimento, ‘‘Influence of human personality in software engineering—A systematic literature review,’’ in *Proc. Int. Conf. Enterprise Inf. Syst. (ICEIS)*, Mar. 2017, pp. 53–62.

[19] R. H. Rutherfoord, ‘‘Using personality inventories to help form teams for software engineering class projects,’’ in *Proc. Annu. Conf. Innov. Technol. Comput. Sci. Educ. (ITiCSE)*, Sep. 2001, pp. 73–76.

[20] N. Gorla and Y. W. Lam, ‘‘Who should work with whom? Building effective software project teams,’’ *Commun. ACM*, vol. 47, no. 6, pp. 79–82, Jun. 2004.

[21] J. Karn and T. Cowling, ‘‘A follow up study of the effect of personality on the performance of software engineering teams,’’ in *Proc. ACM/IEEE Int. Symp. Empirical Soft. Eng.*, Sep. 2006, pp. 232–241.

[22] J. Karn, S. Syed-Abdullah, A. J. Cowling, and M. Holcombe, ‘‘A study into the effects of personality type and methodology on cohesion in software engineering teams,’’ *Behav. Inf. Technol.*, vol. 26, no. 2, pp. 99–111, Mar. 2007.

[23] L. F. Capretz, ‘‘Personality types in software engineering,’’ *Int. J. Human-Comput. Stud.*, vol. 58, no. 2, pp. 207–214, Feb. 2003.

[24] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson, ‘‘Links between the personalities, views and attitudes of software engineers,’’ *Inf. Softw. Technol.*, vol. 52, no. 6, pp. 611–624, Jun. 2010.

[25] M. O. Yilmaz, R. V. O'Connor, R. Colomo-Palacios, and P. Clarke, ‘‘An examination of personality traits and how they impact on software development teams,’’ *Inf. Softw. Technol.*, vol. 86, no. 1, pp. 101–122, Jun. 2017.

[26] P. Sfetsos, I. Stamelos, L. Angelis, and I. Deligiannis, ‘‘An experimental investigation of personality types impact on pair effectiveness in pair programming,’’ *Empirical Soft. Eng.*, vol. 14, no. 1, pp. 187–226, Apr. 2009.

[27] T. Walle and J. E. Hannay, ‘‘Personality and the nature of collaboration in pair programming,’’ in *Proc. Int. Symp. Empirical Soft. Eng. Meas.*, Oct. 2009, pp. 203–213.

[28] J. Chao and G. Atli, ‘‘Critical personality traits in successful pair programming,’’ in *Proc. Conf. AGILE*, Jul. 2006, pp. 89–93.

[29] N. Salleh, E. Mendes, J. Grundy, and G. S. J. Burch, ‘‘An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model,’’ in *Proc. ACM/IEEE Int. Conf. Soft. Eng.*, May 2010, pp. 577–586.

[30] N. Salleh, E. Mendes, and J. Grundy, ‘‘The effects of openness to experience on pair programming in a higher education context,’’ in *Proc. Int. Conf. Soft. Eng. Educ. Training*, May 2011, pp. 149–158.

[31] A. R. Peslak, ‘‘The impact of personality on information technology team projects,’’ in *Proc. ACM SIGMIS-CPR*, Apr. 2006, pp. 273–279.

[32] S. T. Acuña, M. N. Gómez, and N. Juristo, ‘‘How do personality, team processes and task characteristics relate to job satisfaction and software quality?’’ *Inf. Soft. Technol.*, vol. 51, no. 1, pp. 627–639, Mar. 2009.

[33] S. T. Acuña, M. N. Gómez, J. E. Hannay, N. Juristo, and D. Pfahl, ‘‘Are team personality and climate related to satisfaction and software quality? aggregating results from a twice replicated experiment,’’ *Inf. Soft. Technol.*, vol. 57, no. 1, pp. 141–156, Jan. 2015.

[34] M. Shameem, C. Kumar, and B. Chandra, ‘‘A proposed framework for effective software team performance: A mapping study between the team members' personality and team climate,’’ in *Proc. Int. Conf. Comput., Commun. Autom.*, May 2017, pp. 912–917.

[35] A. R. Gilal, J. Jaafar, L. F. Capretz, M. Omar, S. Basri, and I. A. Aziz, ‘‘Finding an effective classification technique to develop a software team composition model,’’ *J. Softw. E*, vol. 30, no. 10, p. e1920, Jan. 2017.

[36] R. Poonam and C. M. Yasser, ‘‘An experimental study to investigate personality traits on pair programming efficiency in extreme programming,’’ in *Proc. Int. Conf. Ind. Eng. Appl.*, Apr. 2018, pp. 96–99.

[37] L. R. Goldberg, ‘‘The international personality item pool and the future of public-domain personality measures,’’ *J. Res. Personality*, vol. 40, no. 1, pp. 84–96, Feb. 2006.

[38] P. N. Tang, M. Steinbatch, and V. Kumar, *Introduction to Data Mining*. Boston, MA, USA: Addison-Wesley, 2006.

[39] J. Han and M. Kamber, *Data Mining, Concepts and Techniques*. Boston, MA, USA: Elsevier, 2006.

[40] D. Galin, *Software Quality Assurance, from Theory to Implementation*. Boston, MA, USA: Addison Wesley, 2004.

[41] T. J. McCabe, ‘‘A complexity measure,’’ *IEEE Trans. Soft. Eng.*, vol. SE-2, no. 4, pp. 308–320, 1976.

[42] K. D. Welker, ‘‘The software maintainability index revisited,’’ *CrossTalk J. Defense Soft. Eng.*, vol. 1, pp. 18–21, Jul. 2001.

[43] *PhpMetrics*. Accessed: Mar. 27, 2019. [Online]. Available: http://www.phpmetrics.org/documentation/index.html

[44] *A Brief Introduction to Function Point*, Accessed: Mar. 27, 2019. [Online]. Available: https://www.cs.helsinki.fi/u/taina/ohtu/fp.html

[45] E. T. Berkman and S. P. Reise, *A Conceptual Guide to Statistics Using SPSS*. Thousand Oaks, CA, USA: SAGE Publications, 2012.

[46] G. A. Morgan, *SPSS for Introductory Statistics*. New Jersey, NJ, USA: Lawrence Erlbaum Associates, Inc. 2004.

[47] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, and M. Oiva, ‘‘Empirical software engineering experts on the use of students and professionals in experiments,’’ *J. Empirical Soft. Eng.*, vol. 23, no. 1, pp. 452–489, Feb. 2018.

[48] S. L. Pfleeger, ‘‘Experimental design and analysis in software engineering,’’ *Ann. Softw. Eng.*, vol. 1, no. 1, pp. 219–253, 1995.

[49] D. I. K. Sjoeberg, "A survey of controlled experiments in software engineering," *IEEE Trans. Softw. Eng.*, vol. 31, no. 9, pp. 733–753, Sep. 2005.
[50] K. Hornbæk, "Some whys and hows of experiments in human-computer interaction," *Found. Trends Hum. Comput. Interact.*, vol. 5, no. 4, pp. 299–373, Jun. 2013.
[51] *Random.Org*. Accessed: Mar. 28, 2018. [Online]. Available: https://www.random.org/
[52] *Personality Test based on Five Factor Model*. Accessed: Feb. 22, 2019. [Online]. Available: http://zelogix.com/personality/
[53] *Team Homogeneity Index (THI) Calculator*. Accessed: Feb. 23, 2019. [Online]. Available: http://zelogix.com/personality/admin/
[54] *Time Keeper*. Accessed: Mar. 10, 2019. [Online]. Available: http://zelogix.com/personality/timelog/
[55] *Test Case Template*. Accessed: Mar. 27, 2019. [Online]. Available: http://zelogix.com/personality/test-case-template.pdf/

NOSHEEN QAMAR is currently pursuing the Ph.D. degree with the National University of Computer and Emerging Sciences (FAST-NUCES). She was a Software Engineer, in 2006, and gained seven years of industry experience with last job title as a Software Development Manager. She is currently an Assistant Professor with The University of Lahore, since February 2015. She did the MS(SPM) from FAST-NUCES, with second position, in 2014. Her research interests include software engineering, design patterns, software teams, project management, and requirements engineering.

ALI AFZAL MALIK received the M.S. and Ph.D. degrees in computer science from the University of Southern California (USC), Los Angeles, USA, in 2007 and 2010, respectively, after receiving the prestigious Fulbright scholarship, in 2005. He was a Software Engineer with Techlogix, in 2003, a well-reputed Pakistani software house. He held an adjunct faculty position at the Lahore University of Management Sciences (LUMS) and a full-time faculty position at the University of Central Punjab (UCP), before joining the National University of Computer and Emerging Sciences (FAST-NUCES), in 2013. He is currently an Assistant Professor and the Head of the Computer Science Department, FAST-NUCES. His current research interests include empirical software engineering, requirements engineering, and software cost estimation. He received the Office of International Services (OIS) Academic Achievement Award, in 2007 and 2010, during his stay at USC. His research paper on the quantitative aspects of requirements elaboration was given the best paper award in SBES 2008, Sao Paulo, Brazil. He has undertaken research in software cost estimation at two of the world's leading research centers in software engineering, i.e., the USC's Center for Systems and Software Engineering (CSSE) and the Institute of Software, Chinese Academy of Sciences (ISCAS).